

## 12、AD 采集 CC2530 温度串口显示

### 1.实验目的

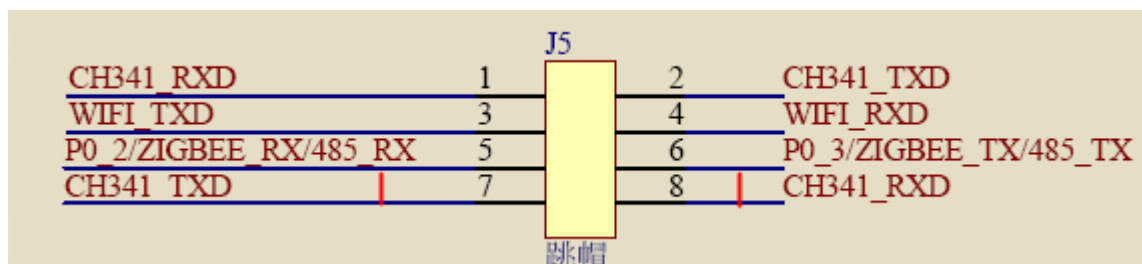
- 1)、通过实验掌握 CC2530 芯片串口配置与使用
- 2)、集到内部温度传感器信息通过串口发送到上位机

### 2.实验设备

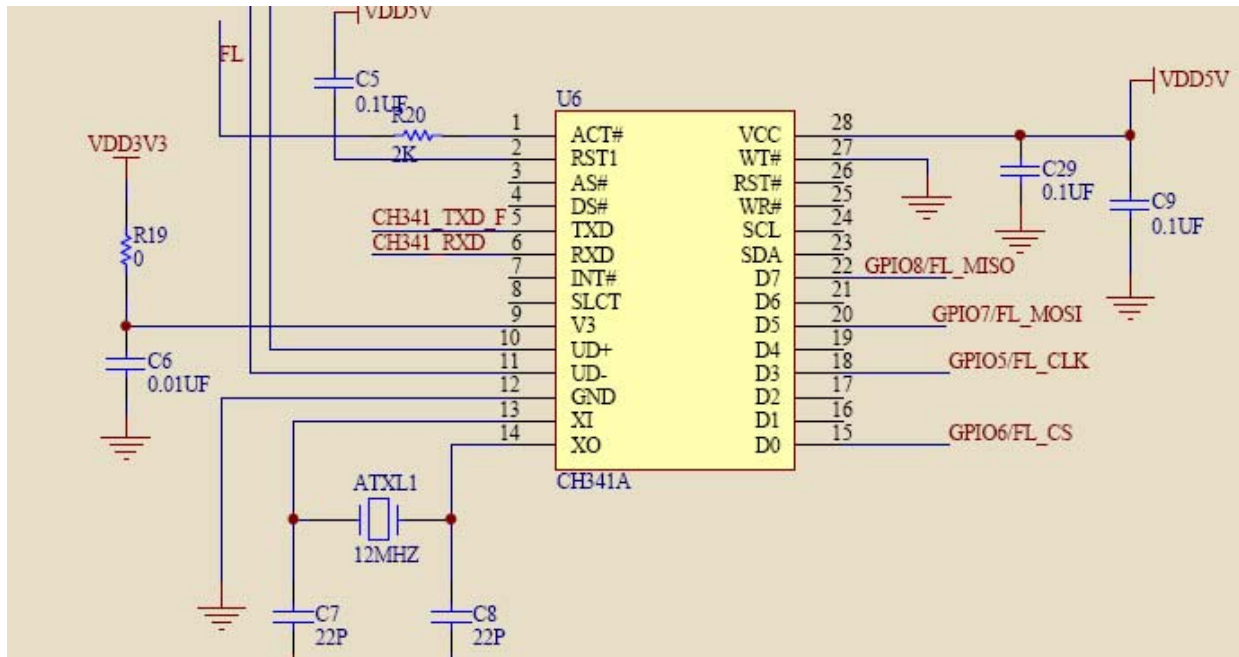
硬件：PC 机一台 ZB 网关（底板、核心板、仿真器、USB 线）一套

软件：2000/XP/win7 系统，IAR 8.10 集成开发环境、串口助手

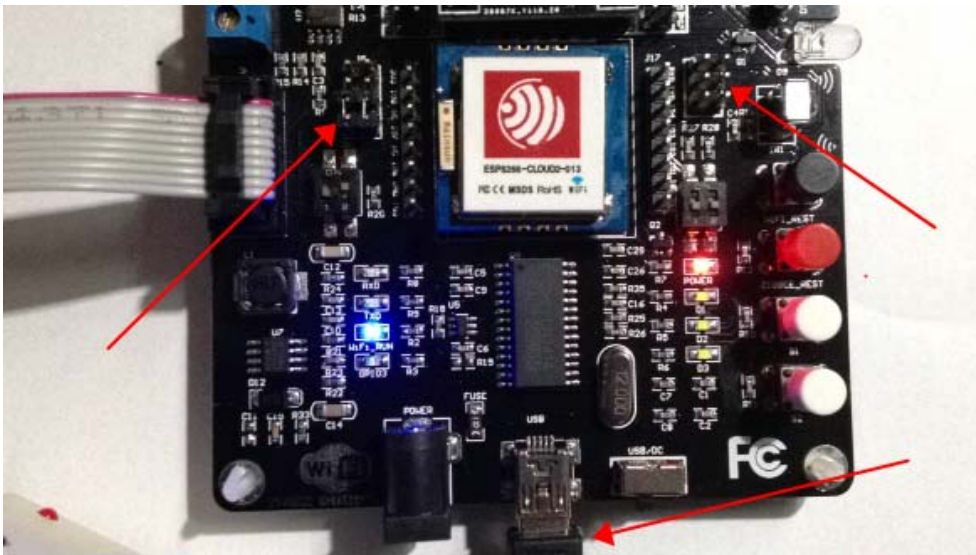
### 3.相关电路图



如上图 J5 这样接，表明 PC 通过 USB 线[板子自带 USB 转串口芯片 CH341]直接接入 ZB 模块，进行对 ZB 的调试使用。如果 3-5 4-6 则表明 ZIGBEE 通过串口控制 wifi 模块。如果 1-3 2-4 则表明 WIFI 模块接入到 PC



如下图就是我们板子实际的短路帽情况：



硬件我们准备好了。我们将板和电脑连接。这个时候需要安装驱动。具体安装方法很简单，下载驱动精灵自动安装。不要自己安装，自己安装容易把驱动搞乱。驱动精灵网址：

<http://pan.baidu.com/s/14fUyU>



P0\_2、P0\_3 配置为外设功能时：P0\_2 为 RX，P0\_3 为 TX. USART0 和 USART1 是串行通信接口，它们能够分别运行于异步 UART 模式或者同步 SPI 模式。两个 USART 具有同样的功能，可以设置在单独的 I/O 引脚。此种串口设计是没有流控功能的。

#### 4. 相关寄存器

寄存器	位	描述
ADCCON1 (0xB4) - ADC 控制1	Bit[7] EOC	转换结束。当ADCH 被读取的时候清除。如果已读取前一数据之前，完成一个新的转换，EOC 位仍然为高。 0: 转换没有完成 1: 转换完成
	Bit[6] ST	开始转换。读为1，直到转换完成 0: 没有转换正在进行 1: 如果ADCCON1.STSEL = 11并且没有序列正在运行就启动一个转换序列。
	Bit[5:4] STSEL	启动选择。选择该事件，将启动一个新的转换序列。 00: P2.0引脚的外部触发。 01: 全速。不等待触发器 10: 定时器1通道0比较事件 11: ADCCON1.ST = 1
	Bit[3:2] RCTRL	控制16 位随机数发生器（第13 章）。当写01 时，当操作完成时设置将自动返回到00。 00: 正常运行。(13X 型展开) 01: LFSR 的时钟一次(没有展开). 10: 保留 11: 停止。关闭随机数发生器
	Bit[1:0]	保留。一直设为11
ADCCON2 (0xB5) - ADC 控制 2	Bit[7:6] SREF	选择参考电压用于序列转换 00: 内部参考电压 01: AIN7 引脚上的外部参考电压 10: AVDD5 引脚 11: AIN6 - AIN7 差分输入外部参考电压
	Bit[5:4] SDIV	为包含在转换序列内的通道设置抽取率。抽取率也决定完成转换需要的时间和分辨率。 00: 64 抽取率(7 位ENOB) 01: 128 抽取率(9 位ENOB) 10: 256 抽取率(10 位ENOB) 11: 512 抽取率(12 位ENOB)

	Bit[3:0] SCH	<p>序列通道选择。选择序列结束。一个序列可以从AIN0 到 AIN7 (SCH&lt;=7) 也可以从差分输入 AIN0-AIN1 到 AIN6-AIN7 (8&lt;=SCH&lt;=11)。对于其他的设置，只能执行单个转换。当读取的时候，这些位将代表有转换进行的通道号码。</p> <p>0000: AIN0 0001: AIN1 0010: AIN2          0011: AIN3 0100: AIN4 0101: AIN5          0110: AIN6 0111: AIN7          1000: AIN0-AIN1 1001: AIN2-AIN3          1010: AIN4-AIN5 1011: AIN6-AIN7          1100: GND 1101: 正电压参考          1110: 温度传感器 1111: VDD/3</p>
ADCCON3 (0xB6) - ADC 控制3	和ADCCON2基本相同，Bit[3:0]有点差异	<p>Bit[3:0] 单个通道选择。选择写ADCCON3 触发的单个转换所在的通道号码。</p> <p>当单个转换完，该位自动清除。</p>
TRO (0x624B) - 测试寄存器0	Bit[0]	<p>设置为1来连接温度传感器到SOC_ADC。也可参见ATEST寄存器描述来使能</p> <p>19.15.3节的温度传感器</p>
ATEST (0x61BD) - 模拟测试控制	Bit[5:0]	<p>控制模拟测试模式：</p> <p>00 0001: 使能温度传感器（也可见12.2.10 节TRO寄存器描述）。其他值保留。</p>
CLKCONCMD 时钟控制命令	Bit[7] OSC32K	<p>32 kHz 时钟振荡器选择</p> <p>0 : 32 kHz XOSC 1 : 32 kHz RCOSC</p>
	Bit[6] OSC	<p>系统时钟源选择</p> <p>0 : 32 MHz XOSC1 : 16 MHz RCOSC</p>
	Bit[5:3] TICKSPD	<p>定时器标记输出设置</p> <p>000 : 32 MHz 001 : 16 MHz 010 : 8 MHz          011 : 4 MHz 100 : 2 MHz 101 : 1 MHz          110 : 500 kHz 111 : 250 kHz</p>
	Bit[2:0] CLKSPD	<p>时钟速度</p> <p>000 : 32 MHz 001 : 16 MHz 010 : 8 MHz          011 : 4 MHz 100 : 2 MHz 101 : 1 MHz          110 : 500 kHz 111 : 250 kHz</p>
CLKCONSTA		<p>CLKCONSTA寄存器是一个只读寄存器, 用来获得当前时钟状态</p>



CC2530串口配置前面章节已详细分析，这里就不再重复了，直接看源码注释更清楚。

温度传感器配置：

```
TR0=0x01;           //设置为1来连接温度传感器到SOC_ADC
```

```
ATEST=0x01;         //使能温度传感
```

AD传感器配置：

```
ADCCON3 = (0x3E);    //选择1.25V为参考电压；14位分辨率；对片内温度传感器采样
```

```
ADCCON1 |= 0x30;     //选择ADC的启动模式为手动
```

```
ADCCON1 |= 0x40;     //启动 AD 转化
```

## 5.源码分析

```
#include "ioCC2530.h"
```

```
#include "initUART_Timer.h"
```

```
#include "stdio.h"
```

```
#include "string.h"
```

```
#include "LCD.h"
```

INT16 AvgTemp;

```
/******
```

温度传感器初始化函数

```
*****/
```

```
void initTempSensor(void){
```

```
    DISABLE_ALL_INTERRUPTS();    //关闭所有中断
```

```
    InitClock();                 //设置系统主时钟为 32M
```

```
    *((BYTE __xdata*) 0x624B) = 0x01; //开启温度传感器,TR0 的地址为 0x624B
```

```
    *((BYTE __xdata*) 0x61BD) = 0x01; //将温度传感器与 ADC 连接起来,ATEST 的地址为
```

```
0x61BD
```

```
}
/*****
读取温度传感器 AD 值函数
*****/
INT8 getTemperature(void){
    UINT8 i;
    UINT16 AdcValue;
    UINT16 value;

    AdcValue = 0;
    for( i = 0; i < 4; i++ )
    {
        ADC_SINGLE_CONVERSION(ADC_REF_1_25_V | ADC_14_BIT | ADC_TEMP_SENS); // 使用 1.25V 内部电压，12 位分辨率，AD 源为：温度传感器
        ADC_SAMPLE_SINGLE(); //开启单通道 ADC
        while(!ADC_SAMPLE_READY()); //等待 AD 转换完成
        value = ADCL >> 2; //ADCL 寄存器低 2 位无效
        value |= (((UINT16)ADCH) << 6);
        AdcValue += value; //AdcValue 被赋值为 4 次 AD 值之和
    }
    value = AdcValue >> 2; //累加除以 4，得到平均值
    return ADC14_TO_CELSIUS(value); //根据 AD 值，计算出实际的温度
}
/*****
主函数
*****/
void main(void)
{
    char i;
    char TempValue[30]={0};
```

```
InitUART0();           //初始化串口
initTempSensor();      //初始化 ADC
```

```
LCD_Init();//初始化 LCD
LCD_CLS();
LCD_welcome();
```

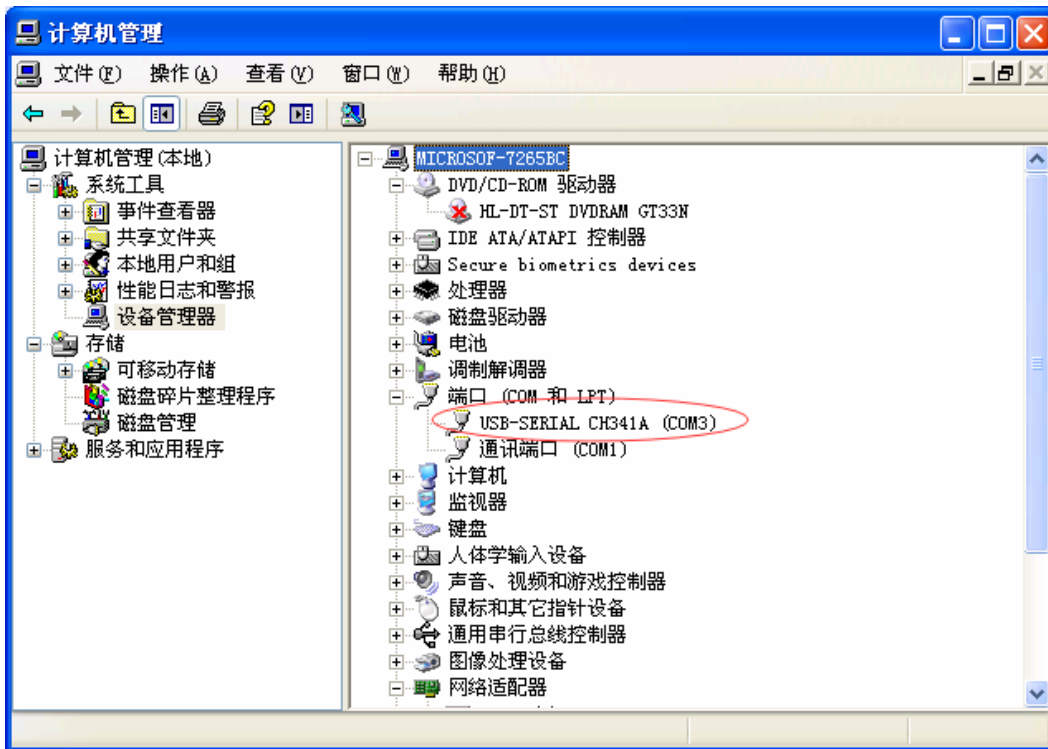
```
while(1)
{
    AvgTemp = 0;
    for(i = 0 ; i < 64 ; i++)
    {
        AvgTemp += getTemperature();
        AvgTemp >>= 1;           //每次累加后除 2.
    }
    memset(TempValue, 0, 30);
    sprintf(TempValue, "ADTestRead = %dC", (INT8)AvgTemp);
    UartTX_Send_String(TempValue,strlen(TempValue));
    UartTX_Send_String("\n",1);
    LCD_P8x16Str(0, 4, (unsigned char*)TempValue);
    Delay(50000);
}
}
```

## 6、实验现象

COM3 是我的 USB 转串口在电脑上生成的，查看方法“我的电脑” - > “设备管理器”，

如  
图：





## 6、实验现象



下面是我们板子的效果哈

